



Modelling and Solving a Practical Flexible Job-Shop Scheduling Problem with Blocking Constraints

Yazid Mati, Chams Lahlou, Stephane Dauzere-Peres

► To cite this version:

Yazid Mati, Chams Lahlou, Stephane Dauzere-Peres. Modelling and Solving a Practical Flexible Job-Shop Scheduling Problem with Blocking Constraints. International Journal of Production Research, 2010, pp.1. 10.1080/00207541003733775 . hal-00592055

HAL Id: hal-00592055

<https://hal.science/hal-00592055>

Submitted on 11 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Modelling and Solving a Practical Flexible Job-Shop Scheduling Problem with Blocking Constraints

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2009-IJPR-0323.R2
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	11-Feb-2010
Complete List of Authors:	Mati, Yazid; Al-Qassim University, College of Business & Economics Lahlou, Chams; Ecole des mines de Nantes, Automatic Control and Industrial Engineering Dauzere-Peres, Stephane; Ecole des Mines de Saint-Etienne, Site Georges Charpak, Centre Microelectronique de Provence
Keywords:	JOB SHOP SCHEDULING, GENETIC ALGORITHMS, DESIGN
Keywords (user):	flexible, blocking



RESEARCH ARTICLE

Modelling and Solving a Practical Flexible Job-Shop Scheduling Problem with Blocking Constraints

Yazid Mati^a, Chams Lahlou^{b*} and Stéphane Dauzère-Péres^c

^a*Al-Qassim University. College of Business & Economics. Almelaida, P.O. Box 6633, Kingdom of Saudi Arabia. Email: matie@qu.edu.sa.*

^b*Ecole des Mines de Nantes - IRCCyN CNRS-UMR 6597. 4, rue A. Kastler. La Chantrerie – BP 20722, F-44307 Nantes Cedex 3 – France.*

^c*Ecole des Mines de Saint-Etienne, Site Georges Charpak – Centre Microélectronique de Provence, 880 avenue de Mimet, F-13541 Gardanne – France. Email: Dauzere-Peres@emse.fr*

(Received 00 Month 200x; final version received 00 Month 200x)

This paper aims at presenting the study of a practical job-shop scheduling problem we modelled and solved when helping a company to design a new production workshop. The main characteristics of the problem are that some resources are flexible, and blocking constraints have to be taken into account. The problem and the motivations for solving it are detailed. The modelling of the problem and the proposed resolution approach, a genetic algorithm, are described. Numerical experiments on real data are presented and analysed. We also show how these results were used to support choices in the design of the workshop.

Keywords: Job-shop scheduling, genetic algorithm, flexible, blocking, design.

*Corresponding author. Email: chams.lahlou@emn.fr

1. Introduction

Designing a manufacturing system involves complex decisions with conflicting objectives such as cost reduction, high equipment utilization and customer satisfaction. One of the important issues is the selection of the proper equipment type and number. In modern manufacturing systems, machines are more and more flexible, and choosing the adequate type is crucial to handle the variety of possible products, demand fluctuations and machine failures. Further, determining the right equipment number helps reducing the procurement and maintenance costs while increasing the equipment utilization.

This paper discusses the problem of selecting machines in a workshop where key components of aircrafts are produced. With the increase of demand and in order to satisfy customers while minimizing costs, the company under consideration was setting a new workshop that could make different part types on the same set of machines. The main challenge the company had to face was the determination of an adequate configuration of the workshop, in order to minimize the cycle times of products. After a first analysis carried out at the company, focusing on demand forecast, a first configuration of the workshop was proposed. In this configuration, products have to go through workstations, and there is one machine in each workstation. The company was interested in evaluating various choices before building the workshop such as: the number of entrant workstations, the functions of some workstations (i.e. the machine type in these workstations), etc.

In the classical literature on equipment selection (see Chen 1999, for instance), aggregate models are proposed and solved in which the details of the actual processing of jobs is not considered. In our approach, and because the scheduling constraints are important, we needed to solve a more detailed model that integrates detailed scheduling features to be sure that the design choices were correct. Our problem is thus formulated as a scheduling problem with several possible scenarios for the design of the workshop. The resulting scheduling problem is a complex shop problem with a material handling system for transporting products, resource flexibility, lack of buffer space between machines, etc.

Flexible shop scheduling problems have attracted many researchers who mainly addressed the job-shop case with multi-purpose machines (also called flexible job-shop), in which an operation may be processed on several machines in a given set, and one of them must be selected. Complexity results are given by Brucker and Schlie (1990), Mati and Xie (2004) for the two-job case. For the flexible job-shop with several jobs, various heuristics are proposed. Most of them use a disjunctive graph for representing feasible solutions and a tabu search heuristic to guide the search. Among these heuristics we may cite the hierarchical approach of Paulli (1995) and the integrated approach of Dautère-Pères and Paulli (1997). A heuristic based on genetic algorithms is developed in Chan *et al.* (2006). This approach is used in an iterative approach to evaluate different flexibility levels of assignment of operations to machines. A generalization to the case where an operation may be processed on several resources simultaneously (multi-resource) is done by Dautère-Pères *et al.* (1998), Mati and Xie (2008a). In the first paper, a tabu search heuristic is proposed using the disjunctive graph representation. The second paper presents a greedy heuristic that schedules jobs one after another according to a job sequence. A genetic algorithm is used to optimize the job sequence.

The blocking constraint arises in production environment where buffer capacity between workstations is limited. Scheduling problems that take into account this constraint are not abundant and the majority of existing studies concentrates on the flow-shop and hybrid flow-shop scheduling problems. A survey on blocking con-

1 constraints is proposed by Hall and Sriskandarajah (1996). For the flow-shop problem,
2 Grabowski and Pempera (2007) use the disjunctive graph model for modelling and
3 propose tabu search heuristics for solving the problem. A genetic algorithm is pro-
4 posed by Caraffa *et al.* (2001) for minimizing the makespan. A branch and bound
5 method is proposed by Ronconi (2008) and some complexity results are given by
6 Martinez *et al.* (2006) for a new type of blocking constraint. For the hybrid flow-
7 shop problem, some heuristic methods are proposed by Wang and Tang (2008),
8 Wardono and Fathi (2004) and exact methods are investigated by Sawik (1995,
9 2000).

10
11 Considering blocking constraints in job-shop environments leads to a control
12 problem called deadlock situation. A system deadlock is a situation that arises when
13 the flow of products is permanently inhibited and/or operations on products cannot
14 be performed. For example, in a robotic cell, a deadlock arises if the robot carries
15 a part to a machine which is working on another part. Under such a situation, the
16 robot cannot unload the part since the machine is not free and needs the robot to
17 download the part it is working on.

18
19 Various solutions have been proposed to handle deadlock situations in auto-
20 mated manufacturing systems (Chu and Xie 1997): deadlock prevention, detection
21 and avoidance methods. The majority of previous studies that deal with schedul-
22 ing and deadlock issues uses Petri nets and reachability graphs to obtain optimal
23 solutions (Lee and DiCesare 1994, ElMaraghy and ElMekawy 2002). Mati *et al.*
24 (2001a,b) propose two heuristics for solving the job-shop scheduling problem with
25 multi-resource operations. The first heuristic uses a greedy heuristic and genetic al-
26 gorithms. The second heuristic makes use of the disjunctive graph and tabu search.
27 Integer linear programming models are proposed in Gomes *et al.* (2005) to schedule
28 flexible job shops under limited intermediate buffers. These models are validated on
29 realistic examples using commercial mathematical programming software. Various
30 characteristics make our problem more complex. A deadlock-free reactive schedul-
31 ing approach is proposed in Fahmy *et al.* (2009), which aims at answering in real
32 time to the occurrence of a variety of possible disruptions in flexible job shops.

33
34 This paper formulates a practical equipment selection problem as a complex
35 scheduling problem and presents a heuristic approach for solving it. The rest of the
36 paper is organized as follows. Section 2 gives a detailed description of the problem
37 encountered by the company, in particular the features of the production environ-
38 nment. Section 3 presents how the problem was modelled, with the assumptions
39 made together with the company. Our resolution method is then described in Sec-
40 tion 4. Numerical experiments are shown in Section 5, together with the analysis
41 that was performed. Finally, Section 6 ends the paper with some conclusions and
42 perspectives.
43
44
45

46 2. Description of the problem

47
48 The problem of selecting equipment for designing a production workshop is for-
49 mulated as a scheduling problem that is solved for each design scenario of the
50 workshop. This approach is possible because the number of alternatives is lim-
51 ited. This section describes the various features of the production environment and
52 the connection between the resulting scheduling problem and those studied in the
53 scheduling literature.
54

55 The company has a set of N products (8 in the specific setting we are con-
56 sidering) $J = \{J_1, J_2, \dots, J_N\}$ that must be performed in the planning horizon
57 H . Each product J_i must be performed in a number of units that differs from
58 one product to another. There are 8 machines (i.e. workstations) in the workshop
59
60

$M = \{M_1, M_2, \dots, M_8\}$, where machines M_1 and M_3 (resp. M_2 and M_4) are of the same type (i.e. can process the same set of operations). Each machine can perform no more than one product at a time and is assumed to be continuously available during the planning horizon. Five different tasks $T \in \{T_1, \dots, T_5\}$ are performed in the production workshop. Task T_1 can be carried out by machines M_2 or M_4 . Task T_2 can be performed by any machine in the set $\{M_1, M_3, M_5\}$, where M_5 is an optional machine that the company could acquire or not. Tasks T_3 , T_4 and T_5 are performed on machines M_6 , M_7 and M_8 , respectively. The layout of the workshop is described in Figure 1.

The products enter the workshop via a loading workstation (called entrant workstation in this paper) where they are put on pallets. Task T_1 is the first operation to be performed for each product. The remaining sequences of operations slightly vary from one product type to another, but the estimated processing times can be very different. Some tasks are not needed for some products and others are repeated several times. The sequence of tasks is $(T_1, T_2, T_3, T_2, T_1, T_2, T_3, T_4, T_5)$ for products J_1 , J_2 and J_3 , $(T_1, T_2, T_1, T_2, T_3, T_4, T_5)$ for product J_4 , and $(T_1, T_2, T_1, T_2, T_4, T_5)$ for products J_5 through J_8 .

Due to the size and weight of the products, there is no buffer space for storing products between machines. Thus, if a product finishes its processing on a machine and the machine on which the product has to be processed next is occupied, then the product has to wait on the current machine, thus blocking it for performing other products. This feature is referred in the scheduling literature to as the blocking constraint (Hall and Sriskandarajah 1996). It is also called the hold-while-wait constraint by authors who address the scheduling and deadlock avoidance problems (Mati et al. 2001b).

With the above features, and according to the scheduling literature, the problem can be seen as a flexible job-shop scheduling problem with blocking constraints, for which some operations have several possibilities to be performed and the repetition of operations is allowed, i.e. the same operation may appear several times in the sequence of a product. An important feature that makes this scheduling problem different from other problems with blocking investigated in the literature is that poor sequencing of machines may lead to undesirable deadlocks (Mati et al. 2001a,b). More precisely, let us consider two products P_1 and P_2 with two operations each. The first operation of P_1 can be performed by M_1 or M_2 and the second operation by M_3 . The first operation of P_2 is performed by M_3 and the second by M_2 . Thus, if one decides to assign M_2 to the first operation of P_1 and to allow the first operations of the two products to be performed simultaneously, then a deadlock situation occurs since P_1 cannot move to its second operation (M_3 is occupied by P_2) and P_2 cannot progress to its second operation (M_2 is occupied by P_1).

Moreover, the manufacturing environment is characterized by the following features that lead to an even more complex scheduling problem.

- *Limited number of pallets.* Products to be produced are put on pallets at the loading (or entrant) workstation. After having visited all necessary machines in the manufacturing process, a product is removed from the pallet on which it is attached, and the pallet is moved to the entrant workstation for processing a new product. Due to the cost of a pallet, only a limited number is allowed limiting the number of products in the workshop at the same time.
- *Material handling.* The workstations are linked by an Automated Material Handling System (AMHS) that moves the products between the machines, and between the machines and the loading/unloading (or entrant) workstations. The AMHS can handle only one product at a time. When the AMHS moves a product

on the machine on which it is processed next, this machine must be available. This is a no-wait constraint that prevents the AMHS to hold products.

- *Storage on machines.* If the machine that processes the next operation of a product is occupied then the product can be stored on any empty machine. This alternative minimizes the blocking times and improves the utilization of critical machines.
- *Negative time lags.* Some operations can be started even though the preceding operation on the routing is not yet completed. This time corresponds to the preparation of the machine by an operator, and represents about 5% of the processing time.

The company was interested in evaluating various choices that had to be made before building the workshop. The first choice concerns the number of entrant workstations from which products are brought into the workshop. Since the processing of products begins with task T_1 , it could be possible, at an additional investment cost, to also use M_4 as an entrant workstation, instead of only M_2 . The second choice concerns the need of purchasing a new workstation (i.e. a new machine M_5) and thus changing the functions of some workstations by modifying manufacturing processes. The last choice concerns the determination of the right number of pallets to be used in the workshop. A smaller number of pallets means less products in the workshop, which may decrease the utilization of machines and therefore increases the total time to process all products. On the other hand, more pallets may lead to more products in the workshop and thus increased congestion due to blocking constraints.

We shall show in the next sections how we modelled and solved the scheduling problem to help the company answer these three questions.

3. Modelling the problem

Taking into account all the features of the production environment results in a complex scheduling model. In order to solve the problem, some constraints were relaxed, validated by the company. This section first discusses how the constraints are modelled and then specifies those that are relaxed. The manufacturing processes of jobs are modified. In what follows, we use the following notation to describe an operation $O_{i,j}$ of a product J_i : $(S_1, \dots, S_t, p_{i,j})$, where $S_k \subset M$ is the k^{th} set of resources that can be used to perform $O_{i,j}$, and $p_{i,j}$ is the processing time which does not depend on the chosen set of resources. As it will be shown below, the resources in our problem are the machines, the Automated Material Handling System (AMHS) and the pallets.

Let us illustrate our modelling choices in the following subsections on two consecutive operations $O_{i,j}$ and $O_{i,j+1}$ of a product J_i described by the routing (1):

$$J_i \equiv \dots; (\{M_1\}, p_{i,j}); (\{M_2\}, \{M_3\}, p_{i,j+1}); \dots \quad (1)$$

This means that, at operation $O_{i,j}$, the product J_i must be processed on machine M_1 with a processing time of $p_{i,j}$ and, at operation $O_{i,j+1}$, can be processed either on machine M_2 or on machine M_3 with a processing time of $p_{i,j+1}$.

3.1 Storage on machines

To integrate the possibility of storing J_i between $O_{i,j}$ and $O_{i,j+1}$ on machines M_4 or M_5 , a waiting operation $O_{i,j+} = (\{M_1\}, \{M_4\}, \{M_5\}, 0)$ is added between $O_{i,j}$ and $O_{i,j+1}$. Machines M_1 is integrated in the set of alternative machines to express

the fact that J_i can also wait on this machine. If one of the machines that can perform $O_{i,j+1}$ is empty, then J_i can move to $O_{i,j+1}$ by advancing instantaneously to $O_{i,j+}$ on machine M_1 and then to $O_{i,j+1}$. The new manufacturing process of J_i is given by the routing (2):

$$J_i \equiv \dots; (\{M_1\}, p_{i,j}); (\{M_1\}, \{M_4\}, \{M_5\}, 0); (\{M_2\}, \{M_3\}, p_{i,j+1}); \dots \quad (2)$$

3.2 Material handling

The AMHS can be explicitly integrated by adding a transportation operation between any two consecutive operations. The added operation (noted j^+ in the example below) needs the AMHS and the processing time corresponds to the time needed for moving products. Adding this information to the manufacturing process of job J_i leads to the one given by the routing (3) where A denotes the AMHS:

$$J_i \equiv \dots; (\{M_1\}, p_{i,j}); (\{A\}, t_{j,j^+}); (\{M_1\}, \{M_4\}, \{M_5\}, 0); (\{A\}, t_{j^+,j+1}); \dots \quad (3)$$

where $t_{j,k}$ is the time needed for the AMHS to bring product J_i from the workstation selected for operation $O_{i,j}$ to the workstation selected for operation $O_{i,k}$. This time depends on the distance between these workstations in the workshop.

3.3 Pallets

Up to this point, every operation needs only one resource to be processed. However, it is necessary to take into account the fact that a pallet is needed for each product from the beginning until the end of its process. Hence, the pallets are modelled as a resource R which is available in a given number of units. The manufacturing process is then modelled by the routing (4):

$$J_i \equiv \dots; (\{M_1, R\}, p_{i,j}); (\{A, R\}, t_{j,j^+}); (\{M_1, R\}, \{M_4, R\}, \{M_5, R\}, 0); \dots \quad (4)$$

The fact that the pallet moves to the entrant workstation after completing the product is modelled by adding an operation at the end of the routing of each product. This operation needs the pallet and the AMHS with a processing time equal to the time needed to move to the entrant workstation.

Note that if the number of pallets is equal to one, the scheduling problem becomes trivial and the solution consists of bringing products one after another in the workshop. If the number of pallets is equal to the number of machines, the problem is equivalent to a flexible job-shop scheduling problem with blocking and thus pallets can be omitted.

3.4 The relaxed constraints

The no-wait constraint that prevents the AMHS to hold the products is always fulfilled since the schedules generated by the solving method are feasible. Otherwise there may be a case where the AMHS moves a product from a given machine M_i and finds the next machine M_j already occupied. This is a deadlock situation since M_j can only be released by the AMHS.

The first constraint that is relaxed in our modelling and solving method is the transportation time that depends on the distance between machines. We use an average time of the transportation times between all workstations of the workshop. After solving the scheduling problem, we recalculate the starting and finishing times of operations by integrating the actual values of the transportation times. Of

course, if a waiting operation uses the same previous machine in the manufacturing process (for instance M_1 in the routing (4)) then the moving time is equal to zero.

The second relaxed constraint is the time lags between two consecutive operations on the routing. We solve the problem without these times and, after finding a solution, a shifting procedure is called to find the actual starting and finishing times of operations.

The third relaxed constraint is related to the limited number of pallets n_p . Integrating this feature in our scheduling model is difficult for the resolution method of (Mati and Xie 2008b), which will be used for evaluating the fitness of chromosomes in the proposed algorithm. One way would be to model each pallet as a separate resource and to integrate flexibility each time an operation needs a pallet. This solution is not convenient as it leads to a huge number of alternative sets (Dautère-Pérès and Paulli 1997). In our case, we assume that the number of pallets is infinite and, during the search process, we save the best schedule with a given maximum number of pallets ($n_p \in \{3, 4, 5, 6, 7\}$ in our numerical experiments). The number of pallets required for a given schedule is determined by finding, over the entire schedule, the maximum number of jobs progressing in parallel in the workshop.

With the above modelling and simplification in mind, the resulting scheduling problem that is solved is a multi-resource job-shop scheduling problem with resource flexibility and blocking, in which an operation may need more than one resource. The AMHS is modelled as a real resource that must be taken into account when building a schedule.

4. Solving the problem

One way for tackling the problem is to model it as a mixed integer linear programming model and to solve it using a commercial solver. Computational experiments carried out by Mati and Xie (2008b) show that commercial solvers are not able to provide reasonably good solutions for the flexible job-shop scheduling problem with blocking constraints, even for instances with less than 10 jobs and 5 machines.

The method used in this paper to answer the questions of the company is an efficient genetic algorithm that uses the greedy procedure proposed by Mati and Xie (2008b) for evaluating the fitness of chromosomes. In this section, genetic algorithms are first briefly described. Then the basic parameters of our genetic algorithm are discussed, including the crossover and mutation operators and the strategy for population management. The method used for evaluating the fitness of the chromosomes is also recalled.

4.1 A brief description of genetic algorithms

Genetic algorithms are population-based procedures that start from an initial population containing a set of feasible solutions represented by chromosomes. The first question to answer when using genetic algorithms is the chromosome encoding (or encoding scheme). There are two types of encoding, the direct and indirect encoding, and the selection of the suitable type depends on the characteristics of the problem. In the direct encoding, a chromosome describes a solution whereas, in the indirect encoding, the chromosome contains information that is used to build a solution. Genetic algorithms iteratively apply genetic operators to replace the current population by a new and hopefully improved population. The first operator is the crossover operator that combines the characteristics of two selected chromo-

somes to generate one or several offsprings. Chromosomes may also be submitted to a mutation operator that consists of a small perturbation of the genes. The purpose of mutation is to prevent the search process from being trapped in local optima if chromosomes in the population become too similar. Genetic algorithms also require a suitable mechanism for population management that decides which chromosomes are selected at each generation for applying the crossover operator. It also requires a fitness function that represents a measure of the quality of each chromosome.

To describe our genetic algorithm, we present below the different parameters of the algorithm, which are fairly standard except the fitness evaluation function that is based on a greedy procedure proposed by Mati and Xie (2008b).

4.2 Parameters of our genetic algorithm

Chromosome encoding. In order to exploit standard crossover operators and to keep the feasibility of the solutions in the reproduction process, we use an indirect encoding in which a chromosome contains an n -string corresponding to a permutation of all integers from 1 to n . Each gene in the chromosome represents a job (product). **The chromosome is then represented by a vector that gives the preference list for the treatment of all jobs. This list is used, by the fitness function described below, to build the actual schedule associated to the chromosome.**

Initial population. In many implementations of genetic algorithms, a good starting population is important for the success of the algorithm. In our case, this population is constructed in a very simple way by randomly generating the chromosomes. **However, we have found that this generally leads to a population with a very poor quality. Since it is not easy to generate a good initial population we have focused on improving it, mainly by increasing the diversity of chromosomes during the population management phase, as explained below.** The size of the population is set to 20 chromosomes. This population size was fixed through numerical experiments conducted on academic test instances.

Chromosome evaluation. A greedy heuristic proposed by Mati and Xie (2008b) is employed to evaluate the chromosomes. This heuristic is recalled below.

Population management. **There are two common strategies to manage the population: population replacement and incremental replacement. In the first case, the new population contains only chromosomes created by crossover and mutations operators, whereas, in the second case, only one chromosome of the parents is replaced with a generated chromosome. The population replacement has been preferred in our application since it allows a high population diversity, allowing the greedy heuristic to start from diverse chromosomes. A large part of the solution space can then be explored and hence the quality of the initial population can be improved.** For selecting pairs of parents to be crossed, the simplest and most used roulette wheel selection is applied, which uses a distribution on the current population that depends on the fitness of the solutions. The probability of each chromosome to be selected is directly proportional to its fitness value.

Crossover and mutation operators. **A general and suitable operator for crossover, in case of representing chromosomes by permutations, is the partially mapped crossover. This crossover has been applied successfully on many scheduling problems, and more specifically on shop problems.** The crossover consists of randomly selecting two cut positions in parents chromo-

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

somes, mapping the portion of one parent located between these positions onto the other parent, and then exchanging the remaining information (see **Goldberg and Lingle (1985)**). After using the crossover operator, a mutation operator is applied that deletes a randomly selected job from its current position in the chromosome and inserts it at another randomly selected position. **One of the advantages of this mutation operator is its good ability to modify the chromosomes and thus to diversify the population.** The crossover operator is applied with a probability equal to 0.9, and the mutation probability is equal to 0.05. These values are standard for scheduling problems and, as for the population size, have been validated on academic test instances.

Stopping criterion. The genetic algorithm terminates after a fixed number of generations. This number has been set to 1,000 through experiments on the industrial data sets. Moreover, if no improvement of the fitness function is done after 200 generations, the algorithm is also stopped.

4.3 Fitness evaluation

The fitness function evaluates the quality of the chromosomes and plays a major role in the selection process. The solution method proposed by Mati and Xie (2008b) is used to measure the fitness of the chromosomes. This method is a greedy heuristic that schedules jobs consecutively according to the job sequence provided by the chromosome. It starts by optimally scheduling the two first jobs using a polynomial algorithm, called *geometric approach*, that reduces the scheduling problem to a shortest path problem in a two-dimensional plane with obstacles. After obtaining the optimal schedule for the two jobs, a *combined job* is defined by combining into a single operation the operations that are performed in parallel. Resource assignment and operation sequencing decisions of scheduled jobs will not be modified. The geometric approach then solves the scheduling problem of the combined job and the next job in the order of the chromosome. The resulting schedule is again used to define a combined job, and the process continues until all jobs are scheduled. The makespan of the final schedule corresponds to the fitness evaluation of the chromosome. For a detailed description of the greedy heuristic for solving shop scheduling problems, the reader is referred to Mati *et al.* (2001a), Mati and Xie (2008a,b).

5. Numerical experiments and analysis

We only present and discuss the results obtained on scenarios with 5 or 6 pallets since the remaining scenarios (with 3, 4 and 7 pallets) give solutions with larger total completion times and waiting times. The computational experiments were conducted on a one-month horizon, during which the workshop must produce in total 30 units of products (see the units per product in Table 1), and 323 hours of work per month (the workshop is open 17 hours per day and 19 days per month). The genetic algorithm was implemented in C++, and the experiments were ran on a Pentium IV (2 GHz). Each scenario took about 10 minutes of CPU time.

The combination of the three parameters (i.e. without or with one new workstation, one or two entrant workstations, five or six pallets) led us to define 8 scenarios. The results for the scenarios are detailed in Tables 2 through 5 and discussed below. For each scenario, the tables show the mean cycle time per product, the total waiting time per machine and the total completion time of all products. The cycle time is the time the unit of a product stayed in the workshop (i.e. completion time

minus arrival time), and the mean cycle time is provided for the number of units of each product.

5.1 *Scenarios with no new workstation*

With one entrant workstation, the total completion time is 374.51 hours (respectively 375.09 hours) if 5 pallets (resp. 6 pallets) are used. The solutions are detailed in Table 2. It is interesting to note that, in this configuration, allowing one more pallet only slightly improves the total waiting time on the machines.

With two entrant workstations, the total completion time is reduced to 367.70 hours (resp. 361.28 hours) if 5 pallets (resp. 6 pallets) are used. As shown in the solutions of Table 3, adding one more pallet improves more the total waiting time than in the previous case. The gain is equal to 3.87 (i.e. 378 - 374.13) in the previous case and equal to 35.79 (i.e. 460.96 - 425.17) in this case.

It must be noted that using two entrant workstations instead of one leads to an improvement of the total completion time (ranging from 7 to 14 hours, i.e. less than one working day), but also to much larger waiting times, in particular when only 5 pallets are used. This can be explained by the fact that, when entrant workstations are not at all bottlenecks, arriving products are quickly pushed forward in the workshop.

5.2 *Scenarios with a new workstation*

With one entrant workstation, the total completion time is 369.75 hours (resp. 354.91 hours) if 5 pallets (resp. 6 pallets) are used. Hence, using 6 pallets is clearly much better and, as shown in Table 4, also leads to drastic improvements of the total waiting time (from 517.35 hours to 430.63 hours).

With two entrant workstations, the total completion time is reduced to 345.38 hours (resp. 332.16 hours) if 5 pallets (resp. 6 pallets) are used. Again, the solutions in Table 5 show that using 6 pallets also improves the total waiting time.

Using two entrant workstations reduces the total waiting time and the total completion time : the gain on the total completion time ranges from 22 hours to 24 hours ($369.75 - 345.38 = 24.37$ hours in case of 5 pallets, and $354.91 - 332.16 = 22.75$ hours in case of 6 pallets), i.e. more than 1 working day. It is interesting to note that, contrary to the case without a new workstation, the total waiting time does not increase with two entrant workstations. This is because the capacity of the workshop has increased, and thus it can handle the products that are sent by the entrant workstations.

5.3 *Answers to the questions of the company*

According to the results presented in the previous subsections, our answers to the questions of the company were the following:

- It is always better to have two entrant workstations, although the improvement is really interesting, given the investment, only if one new workstation is added.
- Adding a new workstation is interesting, but should be coupled with two entrant workstations. The gain (over the scenario with one entrant workstation) on the total completion time is 22.75 hours (with two entrant workstations and 6 pallets), i.e. more than 1.5 days, and 101.18 hours on the total waiting time.
- The workshop should be equipped with 6 pallets with either two entrant workstations or one new workstation.

If one considers the possible configurations, the results of the experiments show that the worst configuration is to use one entrant workstation without a new workstation. An improved configuration is to use either two entrant workstations but no new workstation, or one new workstation but only one entrant workstation. However, in both cases, the investment might be hard to justify given the improvement. Finally, the best configuration is to use two entrant workstations combined with a new workstation.

Finally, the experiments show that 323 working hours per month are not sufficient to handle the production, and hence the company should increase the number of working hours.

6. Conclusion

We have shown how a complex practical scheduling problem could be solved by making some realistic assumptions validated by the company. The various scenarios that have been run helped the company to take important decisions on key features of the new workshop. The next step will probably be, when the workshop is operational, to use the proposed resolution method to design a Decision Support System for its weekly planning. Other characteristics will most likely have to be taken into account, such as the availability of workers.

References

- Akers, S.B., 1956. A graphical approach to production scheduling problems. *Operations Research*, 4, 244–245.
- Brucker, P. and Schlie, R., 1990. Job-shop scheduling with multi-purpose machines. *Computing*, 45, 369–375.
- Caraffa, V., et al., 2001. Minimizing makespan in a blocking flowshop using genetic algorithms. *International Journal of Production Economics*, 70, 101–115.
- Chan, F.T.S., Wong, T.C. and Chan, L.Y., 2006. Flexible job-shop scheduling problem under resource constraints. *International Journal of Production Research*, 44 (11), 2071–2089.
- Chen, M., 1999. A heuristic for solving manufacturing process and equipment selection problems. *International Journal of Production Research*, 37, 359–374.
- Chu, F. and Xie, X.L., 1997. Deadlock analysis of Petri nets using siphons and mathematical programming. *IEEE Transactions on Robotics and Automation*, 13, 793–804.
- Dauzère-Pérès, S. and Paulli, J., 1997. An integrated approach for modelling and solving the multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research*, 70, 281–306.
- Dauzère-Pérès, S., Roux, W. and Lasserre, J.B., 1998. Multi-resource shop scheduling with resource flexibility. *European Journal of Operational Research*, 107, 289–305.
- ElMaraghy, A. H.A. and ElMekkawy, T.Y., 2002. Deadlock-free Scheduling in Flexible Manufacturing Systems Using Petri Nets. *International Journal of Production Research*, 40 (12), 2733–2756.
- Fahmy, S.A., Balakrishnan, S. and ElMekkawy, T.Y., 2009. A generic deadlock-free reactive scheduling approach. *International Journal of Production Research*, 47 (20), 5657–5676.
- Gomes, M.C., Barbosa-Povoa, A.P. and Novais, A.Q., 2005. Optimal scheduling

- for flexible job shop operation. *International Journal of Production Research*, 43 (11), 2323–2353.
- Goldberg, D. and Lingle, R., 1985. Alleles, loci, and the traveling salesman problem. *Proceedings of the 1st international conference on genetic algorithms and their applications*, Lawrence Erlbaum Associates, Publishers, 154–159.
- Grabowski, J. and Pempera, J., 2007. The permutation flow shop problem with blocking: A tabu search approach. *Omega*, 35, 302–311.
- Hall, N. and Sriskandarajah, S., 1996. A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44 (3), 510–525.
- Lee, D.Y. and DiCesare, F., 1994. Scheduling flexible manufacturing systems using Petri nets and heuristic search. *IEEE Transactions on Robotics and Automation*, 10, 123–132.
- Martinez, S. *et al.*, 2006. Complexity of flow-shop with a new blocking constraint. *European Journal of Operational Research*, 169, 855–864.
- Mati, Y., Rezg, N. and Xie, X.L., 2001. Geometric approach and taboo search for scheduling flexible manufacturing systems. *IEEE Transactions on Robotics and Automation*, 17 (6), 805–818.
- Mati, Y., Rezg, N. and Xie, X.L., 2001. A taboo search approach for deadlock-free scheduling of automated manufacturing systems. *Journal of Intelligent Manufacturing*, Special Issue on Metaheuristics, 12 (5/6), 535–552.
- Mati, Y. and Xie, X.L., 2004. The complexity of the two-job shop scheduling problems with multi-purpose unrelated machines. *European Journal of Operational Research*, 152, 159–169.
- Mati, Y. and Xie, X.L., 2008. A genetic search guided greedy algorithm for multi-resource shop scheduling with resource flexibility. *IIE Transactions*, 40 (12), 1228–1240.
- Mati, Y. and Xie, X.L., 2008. Multi-resource shop scheduling with blocking and resource flexibility. *Submitted*.
- Paulli, J., 1995. A hierarchical approach for the FMS scheduling problem. *European Journal of Operational Research*, 86, 32–42.
- Ronconi, D.P., 2005. A branch-and-bound algorithm to minimize the makespan in a flowshop problem with blocking. *Annals of Operations Research*, 138 (1), 53–65.
- Sawik, T.J., 1995. Scheduling flexible flow lines with no in-process buffers. *International Journal of Production Research*, 33 (5), 1357–1367.
- Sawik, T.J., 2000. Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, 31 (13), 39–52.
- Wang, X. and Tang, L., 2009. A tabu search heuristic for the hybrid flowshop scheduling with finite intermediate buffers. *Computers and Operations Research*, 36 (3), 907–918.
- Wardono, B. and Fathi, Y., 2004. A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research*, 155, 380–401.

Tables

Product	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
Units	5	5	1	1	5	5	4	4

Table 1. Number of units per product.

For Peer Review Only

Product	Mean cycle time (hours)		Machine	Waiting time (hours)	
	5 pallets	6 pallets		5 pallets	6 pallets
J_1	59.06	60.73	M_1	49.14	37.76
J_2	55.65	65.06	M_2	68.38	92.59
J_3	77.45	63.59	M_3	18.38	25.68
J_4	43.23	40.67	M_4	64.18	79.41
J_5	35.73	38.78	M_6	27.03	42.12
J_6	31.33	34.68	M_7	36.33	21.89
J_7	82.68	66.59	M_8	114.56	74.68
J_8	37.58	34.95	Total	378.00	374.13
Total completion time (hours)				374.51	375.09

Table 2. Results with no new workstation and one entrant workstation.

Product	Mean cycle time (hours)		Machine	Waiting time (hours)	
	5 pallets	6 pallets		5 pallets	6 pallets
J_1	64.9	62.21	M_1	61.92	60.56
J_2	58.18	62.2	M_2	59.49	72.07
J_3	70.59	73.3	M_3	47.7	10.18
J_4	79.61	59.62	M_4	90.72	90.69
J_5	29.55	41.89	M_6	59	28.92
J_6	43.48	37.83	M_7	31.34	56.54
J_7	70.82	70.83	M_8	110.79	106.21
J_8	35.95	39.17	Total	460.96	425.17
Total completion time (hours)				367.70	361.28

Table 3. Results with no new workstation and two entrant workstations.

For Peer Review Only

Product	Mean cycle time (hours)		Machine	Waiting time (hours)	
	5 pallets	6 pallets		5 pallets	6 pallets
J_1	66.93	66.16	M_1	45	45.21
J_2	65.64	80.78	M_2	110.64	79.7
J_3	65.98	66.9	M_3	17.06	10.78
J_4	47.32	41.81	M_4	95.85	60.35
J_5	37.98	30.51	M_5 (new)	81.33	87.11
J_6	35.06	44.9	M_6	55.1	62.24
J_7	67.17	75.31	M_7	41.82	41.82
J_8	38.29	32.04	M_8	70.55	43.42
Total			Total	517.35	430.63
Total completion time (hours)				369.75	354.91

Table 4. Results with a new workstation and one entrant workstation.

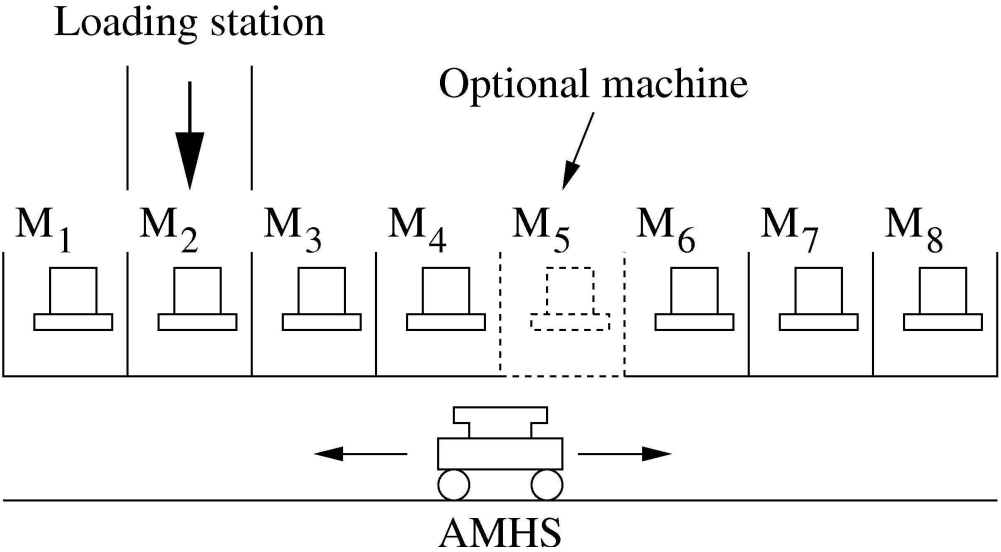
Product	Mean cycle time (hours)		Machine	Waiting time (hours)	
	5 pallets	6 pallets		5 pallets	6 pallets
J_1	60.06	52.7	M_1	14.8	15.35
J_2	61.89	70.37	M_2	46.72	33.39
J_3	61.06	64.81	M_3	16.57	24.38
J_4	55.01	45.51	M_4	82.72	90.88
J_5	30.91	40.61	M_5 (new)	50	68.12
J_6	28.85	34.07	M_6	48.1	23.46
J_7	69.7	64.46	M_7	38.66	19
J_8	38.33	36.06	M_8	70.88	49.41
Total			Total	368.45	323.99
Total completion time (hours)				345.38	332.16

Table 5. Results with a new workstation and two entrant workstations.

List of figure captions

1. Layout of the production workshop.

For Peer Review Only



Layout of the production workshop
102x55mm (600 x 600 DPI)